

**0000000000000000000000000000000000**

[illegible]

## FIELD OF THE INVENTION

**[0001]** This invention generally relates to changing the bandwidth between a device and a processor. More particularly this invention relates to an apparatus and method capable of changing the bandwidth between a device and a processor without changing the physical component layout in the processor.

## BACKGROUND OF THE INVENTION

[0002] Joining several processors in parallel increases processing capacity. Typically, any number from two to eight processors may be joined in parallel. Generally, multiple parallel processors are joined together on a shared bus. Figure 1 illustrates a four processor (4P) architecture used in conjunction with a shared bus. Four processors, Processor 1, Processor 2, Processor 3, and Processor 4, connect to a shared bus, which in turn connects to the Northbridge chipset. The Northbridge chipset further connects to the Southbridge chipset and external memory. For example, a Pentium™ processor may employ the shared bus architecture illustrated in figure 1. However, a point-to point architecture, typically, provides a higher bandwidth than does a shared bus architecture.

**[0003]** In a shared bus architecture, multiple devices all share the same bus and must follow an order and protocol to use the bus. In contrast, a point-to-point bus architecture provides an uninterrupted connection between two separate devices. Thus, in general, a point-to-point bus creates a higher bandwidth between two separate devices. A higher bandwidth can have the



0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99

**[0014]** figure 9 illustrates an embodiment of the routing agent controlling the communication pathways in an outbound information transfer layer.



**7-0-1**

[0017] A routing agent may be used with a processor in a point-to-point architecture to allow a customer to change the bandwidth, or number of bus lines and clock speed, between two devices. A point-to-point bus architecture provides an uninterrupted connection between two separate devices. Generally, a packet based protocol transfers information in a point-to-point bus architecture. The routing agent may alter the effective bandwidth between a processor and a device exterior to the processor by changing one or more communication pathways in the processor without changing a physical component layout in the processor. Bandwidth is the total amount of information that can be transferred within a given time period between two devices. In an embodiment, the manufacturer or the customer may change the number of ports linked between the two devices to increase the bandwidth between the devices. In an embodiment, the manufacturer or the customer may add additional processors linked to a device by changing the number of ports linked between a first processor and the device exterior to the first processor.

















agent **702**, a fifth 128-bit register **724**, a local address transaction tracker buffer (LATT) **726**, a central data management buffer (CDM) **728**, and a remote address transaction tracker buffer (RATT) **730**.

[0029] In an embodiment of the inbound buffer layer **700**, if the configuration register **732** is programmed for a 32-bit point-to-point connection, then the routing agent **702** effectively gangs two link layers together. The first link layer **704** receives the eight related 16-bit packets (128-bit) of information at twice the clock speed from the communication switching device in the first physical layer. The first link layer **704** stores the 128-bit packet of information in the first register **706**. The 128-bit packet of information is routed appropriately to either the first response queue **708** or the first request queue **710**. The routing agent **702** sends an enablement signal to the first communication pathway switching device **712**, third communication pathway switching device **734**, and fifth communication pathway switching device **722**. The routing agent **702** also sends a disable signal to the second communication pathway switching device **716** and fourth communication pathway switching device **736**. The routing agent **702** directs the fifth communication pathway switching device **722** to request data only from the first communication pathway switching device **712** and the third communication pathway switching device **734**. In 32-bit mode, the second communication pathway switching device **716** and fourth communication pathway switching device **736** receive no signal from the physical layer because the routing agent **702** disabled the second flip flop and fourth flip flop in the physical layer.

[0030] The 128-bit packets of information are transferred to the fifth communication pathway switching device **722** through the first communication pathway switching device **712** and then the third communication pathway switching device **734** in a cyclic sequential manner. In a

repeating cycle, the fifth communication pathway switching device **722** requests the contents of the first response queue **708**, then the first request queue **710**, then the third response queue **738**, then the third request queue **740**, and then repeats this sequence. Every packet of information may be either a request for a command and/or data or a response to a command or data. In an embodiment, once a packet is written into a response queue or request queue, then the information is transferred from these queues at core clock frequency. The packets of information are sent from the fifth communication pathway switching device **722** to one of the following three components. The central data management buffer (CDM) **728** stores data to be consumed by the protocol layer **740**. The remote address transaction tracker buffer (RATT) **730** stores commands from a remote device such as a request from a remote processor. The CDM **728** stores corresponding data, if any, associated with that command. Similarly, the local address transaction tracker buffer (LATT) **726** stores local commands and responses made by the processor core. The corresponding data, if any, associated with that command is stored in the CDM **728**.

**[0031]** If the configuration register **732** is programmed for a 16-bit point-to-point connection, then each inbound link layer **704**, **714**, **718**, **720** acts independently. The routing agent **702** directs the fifth communication pathway switching device **722** to request information out of the eight possible sources, the corresponding request queue **710**, **713**, **740**, **750** or the corresponding response queue **708**, **711**, **738**, **748** in each link layer **704**, **714**, **718**, **720**, in a cyclic sequential manner. If a particular queue is empty, then the communication pathway switching device **722** automatically pulls packets out of the next queue without any penalty. In an embodiment, the packets from the four physical layers are written into the appropriate queue at 100 megahertz, if





layer **804** delivers commands generated from itself to the LATT **814** and data, if any, corresponding to that command to the CDM **816**. The protocol layer **804** delivers responses to commands from remote devices to the RATT **818** and data, if any, corresponding to that command to the CDM **816**. If the configuration register **820** is programmed for a 32-bit point-to-point connection, then the routing agent **802** effectively enables two of the four outbound link layers. The following happens in tracing the signal path in the first outbound link layer **806** and the second outbound link layer **808**. The routing agent **802** directs the second communication pathway switching device **822** and the fourth communication pathway switching device (not shown) to be disabled. The routing agent **802** directs the fifth communication pathway switching device **828** to request data through the first communication pathway switching device **824** and third communication pathway switching device **826** from their respective request buffer **830** or response buffer **832**. The fifth communication pathway switching device **828** alternates, in a repeating cycle, requests for bits from the request buffer **830** and the response buffer **832**. The fifth communication pathway switching device **828** also receives any data, if any, corresponding to those commands from the CDM **816**. The fifth communication pathway switching device **828** then routes the 128-bit data packet through a first time delay **833** to either the outbound response queue **836** or outbound request queue **834** located in the outbound physical layer.

[0034] One or more retry queues 838 exist in the outbound link layer. The retry queue 838 stores a duplication of the information to be transferred in case an error occurs in the transfer of the information. The retry queue 838 exists primarily in case there may be an error in the link. The retry queue 838 erases the duplicated information stored in a retry queue 838 upon receiving a confirmation of an error free transfer of information.





between the processor and the input-output component, then routing agent **902** directs each physical layer to act independently of the other physical layers. A fourth communication pathway switching device **926** reads the 32-bit packet of information at 400 megahertz from the third register **920** and outputs two 16-bit packets of information. A fifth communication pathway switching device **928** reads the 16-bit packet of information at 400 megahertz from the forth register **927** and outputs the 16-bit packet of information to the first port **904**. The routing agent **902** sends a signal to the sixth communication pathway switching device **930** in the second outbound physical layer **932**. The signal enables the 16-bit path in the fifth communication pathway switching device **928**. The routing agent **902** disables the 32-bit bypass path **922** from the fourth communication pathway switching device **926**. The routing agent **902** also sends a signal to the second physical layer **932**, the sixth communication pathway switching device **930**, to enable the 16-bit path in the sixth communication pathway switching device **930** and disable/ignore the 32-bit bypass path **922** coming from the third register **920**. Thus, the output from first port **904** and the second port **906** are unrelated 16-bit packets of information.

[0039] If in one embodiment for example, a 32-bit point-to-point connection exist between the processor and the input-output component, then the routing agent **902** gangs the output of the first port **904** and the output of the second port **906**. The routing agent **902** sends a signal to the sixth communication pathway switching device **930**. The routing agent's signal disables the 16-bit path in the sixth communication pathway switching device **930** and enables the 32-bit bypass path **922** coming from the third register **920**. The routing agent **902** also sends a signal to the first physical layer **931**, the fourth communication pathway switching device **926**, to enable the 32-bit bypass path **922**. Thus, the output from first port **904** and the second port **906** are related



[illegible]